



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/726,272	12/01/2003	Satyanarayana Raju	50325-0847	3352

29989 7590 04/05/2007  
HICKMAN PALERMO TRUONG & BECKER, LLP  
2055 GATEWAY PLACE  
SUITE 550  
SAN JOSE, CA 95110

EXAMINER
----------

QIAN, SONGWEI

ART UNIT	PAPER NUMBER
----------	--------------

2109

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	04/05/2007	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

## Office Action Summary

Application No.

10/726,272

Applicant(s)

RAJU ET AL.

Examiner

Songwei Qian

Art Unit

2109

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 01 December, 2003.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-30 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-30 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 01 December 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)            | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)   | Paper No(s)/Mail Date. _____                                      |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date <u>07/19/2004</u> .  | 6) <input type="checkbox"/> Other: _____                          |

***DETAILED ACTION***

1. Claims 1-30 are pending in this application.

***Claim Objections***

2. Claim 9 is objected to because of the following informalities: the phrase "further comprising" is repeated. Appropriate correction is required.

***Claim Rejections - 35 USC § 101***

3. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. Claims 18-23 are rejected under 35 U.S.C. 101 as the claimed invention is directed to non-statutory subject matter.

5. As for claims 18-23, the claims fail to place the invention squarely within one statutory class of invention. In the specification, applicant has provided evidence that applicant intends the "computer-readable medium" to include "a carrier wave" ([0029], Line 5, and other places). As such, the claim is drawn to a form of energy. Energy is not one of the four categories of invention and therefore this claim is not statutory.

Energy is not a series of steps or acts and thus is not a process. Energy is not a physical article or object and as such is not a machine or manufacture. Energy is not a combination of substances and therefor not a composition of matter.

***Claim Rejections - 35 USC § 112***

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claims 2-4, 6, 8-10, 12, 14-15, 17, 19-21, 23, 25, 27-28, and 30 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

8. As for claims 2-4, 9-10, 12, 14-15, 19-21, 25, and 27-28, the phrase "the second software component" is not clearly understood and renders the claims indefinite. It is unclear what the phrase means, the current version of the second software component that needs to be changed as cited in claims 1, 7, 11, 18, and 24, or a new version of the second software component. For examining purpose, the phrase "the second software component" is interpreted as "new version of the second software component that needs to be changed as cited in claims 1, 7, 11, 18, and 24".

9. As for claims 6, 8, 17, 23, and 30, the phrase "the second software component" as listed in the following places, is not clearly understood and renders the claims indefinite. It is unclear what the phrase means, the current version of the second software component that needs to be changed as cited in claims 1, 7, 11, 18, and 24, or a new version of the second software component. For examining purpose, the phrase "the second software component" is interpreted as "new version of the second software component that needs to be changed as cited in claims 1, 7, 11, 18, and 24".

- a) claim 6, Line 2, and second occurrence in Lines 3-4;
- b) claim 8, Line 2, and second occurrence in Lines 3-4;
- c) claim 17, Line 3, and second occurrence in Lines 5-6;
- d) claim 23, Line 4, and second occurrence in Lines 5-6;
- e) claim 30, Line 2, and second occurrence in Lines 4-5.

***Claim Rejections - 35 USC § 103***

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

11. Claims 1-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mayer (US Pub. # 2002/0019864 A1) in view of Erickson et al. (US Pub. # 2003/0177223 A1), hereinafter "Erickson".

12. As for claim 1, Mayer discloses:

A method for managing versions of a plurality of software components on a network ([0017]), comprising:

detecting a version change to a first software component out of the plurality of the software components (determine any configuration changes of the managed elements , [0022], Lines 3-4);

Although Mayer discloses "a database containing the configuration of managed elements" (Fig. 6 and [0043]), Mayer does not appear to explicitly disclose:

automatically identifying a second software component out of the plurality of the software components, that needs to be changed to be compatible with the first software component, wherein the second software component depends on the first software component.

However, Erickson discloses:

automatically identifying a second software component out of the plurality of the software components, that needs to be changed to be compatible with the first

Art Unit: 2109

software component, wherein the second software component depends on the first software component (Fig. 2 and [0019], Lines 1-5; note that "the computing device 110 may include a variety of devices having a programmable device, such as, but not limited to, servers, personal computers, PDAs, etc.", [0024], Lines 23-25).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Mayer with the teachings of Erickson by identifying a second software component out of the plurality of the software components that needs to be changed to be compatible with the first software component, wherein the second software component depends on the first software component, in order to insure that the versions (managed elements) are compatible (Erickson, [0004], Lines 7-8) and in order for the (network) system to function properly (Erickson, [0005], Line 5).

13. As for claim 7, Mayer discloses:

A method for managing versions of a plurality of software components on a network ([0017]), comprising:

detecting a version change to a first software component out of the plurality of the software components (determine any configuration changes of the managed elements , [0022], Lines 3-4);

Although Mayer discloses "a database containing the configuration of managed elements" (Fig. 6 and [0043]), Mayer does not appear to explicitly disclose:

automatically identifying a second software component out of the plurality of the software components that needs to be changed to be compatible with the first software component, wherein the second software component depends on the first software component;

collecting attributes of the second software component; and

automatically manipulating the second software component according to the attributes.

However, Erickson discloses:

automatically identifying a second software component out of the plurality of the software components that needs to be changed to be compatible with the first software component, wherein the second software component depends on the first software component (Fig. 2 and [0019], Lines 1-5; note that "the computing device 110 may include a variety of devices having a programmable device, such as, but not limited to, servers, personal computers, PDAs, etc.", [0024], Lines 23-25);

collecting attributes of the second software component (firmware versions, [0016], Lines 1-2); and

automatically manipulating the second software component according to the attributes ([0027], Lines 7-11).



Art Unit: 2109

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Mayer with the teachings of Erickson in order to insure that the versions (managed elements) are compatible (Erickson, [0004], Lines 7-8), to avoid unnecessary download of software updates, to avoid large network transport overhead (Mayer, [0012], Lines 4-5), to avoid excessive time required to perform software updates by the system administrator (Erickson, [0006], Lines 15-17), and to avoid mistakes made by the system administrator for updates (Erickson, [0006], Lines 11-12), and in order for the (network) system to function properly (Erickson, [0005], Line 5).

14. As for claim 11, Mayer discloses:

An apparatus for managing versions of a plurality of software components on a network ([0017]), comprising:

a user interface (management interface, [0058], Lines 1-2); and

a processing engine (distributed managers 2 and agent means, Fig. 1, [0052], Line 2 and [0028], Line 1), coupled to the user interface, wherein the processing engine further includes:

an event manager (intelligent agents, [0026], Line 5) that detects a version change to a first software component out of the plurality of the software components ([0026], Line 4 and [0022], Lines 3-4);

Mayer does not appear to explicitly disclose:

Art Unit: 2109

a component manager that in response obtains version information of first software component from a version manager and automatically identifies a second software component out of the plurality of the software components that needs to be changed to be compatible with the first software component, wherein the second software component depends on the first software component.

However, Erickson discloses:

a component manager (a network administrator console (NAC) 140 and a service processor 124, Fig. 1, [0015], Line 1 and [0013], Lines 6-7) that in response obtains version information of first software component from a version manager (version check utility 142, Fig. 1 and [0016], Line 1) and automatically identifies a second software component out of the plurality of the software components that needs to be changed to be compatible with the first software component, wherein the second software component depends on the first software component (Fig. 2 and [0019], Lines 1-5; note that "the computing device 110 may include a variety of devices having a programmable device, such as, but not limited to, servers, personal computers, PDAs, etc.", [0024], Lines 23-25).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Mayer with the teachings of Erickson in order to insure that the versions (managed elements) are compatible (Erickson, [0004], Lines 7-8) and in order for the (network) system to function properly (Erickson, [0005], Line 5).

15. As for claim 18, Mayer discloses:

detecting a version change to a first software component out of the plurality of the software components (determine any configuration changes of the managed elements , [0022], Lines 3-4);

Although Mayer discloses “a database containing the configuration of managed elements” (Fig. 6 and [0043]), Mayer does not appear to explicitly disclose:

automatically identifying a second software component out of the plurality of the software components, that needs to be changed to be compatible with the first software component, wherein the second software component depends on the first software component.

However, Erickson discloses:

automatically identifying a second software component out of the plurality of the software components, that needs to be changed to be compatible with the first software component, wherein the second software component depends on the first software component (Fig. 2 and [0019], Lines 1-5; note that “the computing device 110 may include a variety of devices having a programmable device, such as, but not limited to, servers, personal computers, PDAs, etc.”, [0024], Lines 23-25).

Art Unit: 2109

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Mayer with the teachings of Erickson by identifying a second software component out of the plurality of the software components that needs to be changed to be compatible with the first software component, wherein the second software component depends on the first software component, in order to insure that the versions (managed elements) are compatible (Erickson, [0004], Lines 7-8) and in order for the (network) system to function properly (Erickson, [0005], Line 5).

16. As for claim 24, Mayer discloses:

a user interface means (management interface, Fig. 8 and [0058], Lines 1-2);

and

a processing means (distributed managers 2 and agent means, Fig. 1, [0052], Line 2 and [0028], Line 1), coupled to the user interface, wherein the processing means

further includes:

a detection means (distributed managers 2 and agents 3, Fig. 1 and [0052], Line 2) for detecting a version change to a first software component out of the plurality of the software components (determine any configuration changes of the managed elements , [0022], Lines 3-4);

Mayer does not appear to explicitly disclose:

a compatibility verification means for automatically identifying a second software component out of the plurality of the software components that needs

to be changed to be compatible with the first software component, wherein the second software component depends on the first software component.

However, Erickson discloses:

a compatibility verification means (a network administrator console (NAC) 140 and a service processor 124, Fig. 1, [0015], Line 1 and [0013], Lines 6-7) for automatically identifying a second software component out of the plurality of the software components that needs to be changed to be compatible with the first software component, wherein the second software component depends on the first software component (Fig. 2 and [0019], Lines 1-5).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Mayer with the teachings of Erickson by using a compatibility verification means as taught by Erickson to automatically identify a second software component out of the plurality of the software components that needs to be changed to be compatible with the first software component, wherein the second software component depends on the first software component, in order to insure that the versions (managed elements) are compatible (Erickson, [0004], Lines 7-8) and in order for the (network) system to function properly (Erickson, [0005], Line 5).

17. As for claims 2, 12, 19, and 25 Erickson discloses:

automatically downloading the second software component from a network device ([0017], Lines 9-10).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Mayer with the teachings of Erickson by automatically downloading the second software component from a network device as taught by Erickson in order to avoid excessive time required to perform software updates by the system administrator (Erickson, [0006], Lines 15-17) and to avoid mistakes made by the system administrator for updates (Erickson, [0006], Lines 11-12).

18. As for claims 3, 9, 14, 20, and 27, Erickson discloses:

storing a copy of the second software component in a cache (The downloaded version is stored in the memory 120, [0017], Lines 12-13).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Mayer with the teachings of Erickson by storing a copy of the second software component in a cache as taught by Erickson in order to perform software updates.

19. As for claims 4, 10, 15, 21, and 28, Erickson discloses:

checking version information of the second software component that is stored in the cache to determine whether to perform the downloading step ([0017], Lines 1-5).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Mayer with the teachings of Erickson by checking version information of the second software component that is stored in the cache to determine whether to perform the downloading step as taught by Erickson in order to avoid unnecessary download of software updates and to avoid large network transport overhead (Mayer, [0012], Lines 4-5).

20. As for claims 5 and 22, Erickson discloses:

collecting attributes of the second software component (firmware versions, [0016], Lines 1-2); and

automatically manipulating the second software component according to the attributes ([0027], Lines 7-11).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Mayer with the teachings of Erickson in order to avoid unnecessary download of software updates, to avoid large network transport overhead (Mayer, [0012], Lines 4-5), to avoid excessive time required to perform software updates by the system administrator (Erickson, [0006], Lines 15-17), and to avoid mistakes made by the system administrator for updates (Erickson, [0006], Lines 11-12).

Art Unit: 2109

21. As for claims 6, 8, 17, 23, and 30, Erickson discloses:

downloading the second software component as part of performing an instance of the method ([0017], Lines 9-10); and

replacing an existing version of the second software component with the second software component that has been downloaded in the same instance ([0017], Lines 11-12).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Mayer with the teachings of Erickson in order to avoid excessive time required to perform software updates by the system administrator (Erickson, [0006], Lines 15-17), and to avoid mistakes made by the system administrator for updates (Erickson, [0006], Lines 11-12).

22. As for claims 13 and 26, Erickson discloses:

the component manager (a network administrator console (NAC) 140 and a service processor 124, Fig. 1, [0015], Line 1 and [0013], Lines 6-7) informs an operator of the apparatus of the second software component via the user interface (generate an alert (step 350), Fig. 3 and [0026], Line 2).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Mayer with the teachings of Erickson by informing an operator of the apparatus of the second software component via the user



interface as taught by Erickson in order to allow the operator to know which apparatus has a compatibility issue and to allow the operator to address the compatibility issue (Erickson, [0026], Line 9).

23. As for claims 16 and 29, Erickson discloses:

a desktop manager (a network administrator console (NAC) 140, Fig. 1 and [0015], Line 1), coupled to the component manager (a network administrator console (NAC) 140 and a service processor 124, Fig. 1, [0015], Line 1 and [0013], Lines 6-7), wherein the desktop manager

collects attributes of the second software component (firmware versions, [0016], Lines 1-2); and

manipulates the second software component according to the attributes ([0027], Lines 7-11).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Mayer with the teachings of Erickson in order to avoid unnecessary download of software updates, to avoid large network transport overhead (Mayer, [0012], Lines 4-5), to avoid excessive time required to perform software updates by the system administrator (Erickson, [0006], Lines 15-17), and to avoid mistakes made by the system administrator for updates (Erickson, [0006], Lines 11-12).


**Conclusion**

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Songwei Qian whose telephone number is 571-270-1910. The examiner can normally be reached on M-F (alternative Friday off 8:00am thru 5:00pm).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nabil El-Hady can be reached on 571-272-3963. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

SQ  
03/21/2007

  
NABIL M. EL-HADY  
SUPERVISORY PATENT EXAMINER